MedLinker: Medical Entity Linking with Neural Representations and Dictionary Matching

Daniel Loureiro(✉) and Alípio Mário Jorge

LIAAD - INESCTEC, Porto, Portugal
{dloureiro,amjorge}@fc.up.pt

Abstract. Progress in the field of Natural Language Processing (NLP) has been closely followed by applications in the medical domain. Recent advancements in Neural Language Models (NLMs) have transformed the field and are currently motivating numerous works exploring their application in different domains. In this paper, we explore how NLMs can be used for Medical Entity Linking with the recently introduced MedMentions dataset, which presents two major challenges: (1) a large target ontology of over 2M concepts, and (2) low overlap between concepts in train, validation and test sets. We introduce a solution, MedLinker, that addresses these issues by leveraging specialized NLMs with Approximate Dictionary Matching, and show that it performs competitively on semantic type linking, while improving the state-of-the-art on the more fine-grained task of concept linking (+4 F1 on MedMentions main task).

Keywords: Entity Linking · Bioinformatics · Neural Language Models

1 Introduction

Medical Entity Recognition and Linking remain challenging tasks at the intersection between Natural Language Processing (NLP) and Information Retrieval (IR). The main difficulty arises from the fact that annotated datasets are scarce and particularly expensive to collect (require domain expertise), while the ontologies used in this domain are also especially large. From the standpoint of NLP, the relatively small and low-coverage datasets are hard to model using current neural approaches, whereas IR is limited by the subtle semantics underlying the different concepts that constitute the ontologies.

The recently introduced MedMentions [1] dataset provides the largest set of mention-level annotations targeting the UMLS (Unified Medical Language System) ontology. UMLS [6] is a compilation of several medical ontologies, making it the most comprehensive and broad, spanning a range of topics from viruses to

The research leading to these results has received funding from the European Union's Horizon 2020 - The EU Framework Programme for Research and Innovation 2014–2020, under grant agreement No. 733280.

J. M. Jose et al. (Eds.): ECIR 2020, LNCS 12036, pp. 230–237, 2020.
https://doi.org/10.1007/978-3-030-45442-5_29

biomedical occupations. Even though the MedMentions annotation effort was a substantial undertaking, it falls short of covering the full set of concepts comprising UMLS (∼1% coverage), as well as displaying low overlap between the set of concepts occurring in its training splits and the set of concepts occurring in the development and test splits (∼50% overlap). In order to overcome the challenges presented in this dataset, we propose a solution that's based on Neural Language Models (NLMs) but designed to fallback on Approximate Dictionary Matching (ADM) for zero-shot entity linking, taking advantage of the large lexicon provided with the UMLS Metathesaurus. Unlike previous approaches using NLMs in the medical domain [2–5], our solution decouples mention recognition and entity linking, leveraging NLMs for these subtasks in separate modules, and allowing for other methods, namely ADM, to take part in the linking process.

In this work we explore approaches using NLMs for the related task of Word Sense Disambiguation (WSD), particularly pooling methods for representing spans in NLM-space [9]. We show that the Semantic Type (STY) and Concept Unique Identifiers (CUI) embeddings learned in the process are useful for our linking tasks, and can be effectively combined with ADM for improved performance over previous solutions. While our solution, MedLinker[1], is designed for MedMentions, the breadth of the target ontology makes it useful elsewhere.

2 Related Work

In this section we focus on previous works using NLMs for biomedical NER, or addressing MedMentions directly. While there are already several works using the latest Transformer-based NLMs for biomedical tasks [2–4], these have, so far, focused only on the adaptation of the pretrained NLM for the medical domain, without considering how these can be leveraged with complementary approaches.

The authors of MedMentions reported results on a subset of their corpus (st21pv) using a popular method for biomedical NER called TaggerOne [8]. This method learns to jointly predict spans and link entities but relies mostly on discrete features. Also, MedMention's authors claimed that it took them several days to train their model with high-performance computing resources (e.g. 900 GB of RAM), raising tractability concerns about using TaggerOne.

The first results on applying NLMs to MedMentions have been recently reported by [5]. Their solution showed strong performance for semantic type linking, but followed the standard approach for NER tasks also used by [2–4].

3 Data

The MedMentions dataset is provided in two variants, one targeting the full ontology of UMLS, and another targeting a subset of that ontology[2] selected

[1] Package, code, and additional results: https://github.com/danlou/medlinker.
[2] Based on UMLS release 2017 AA Active - num. concepts: 3.4M (full), 2.3M (st21pv).

by domain experts as particularly interesting for medical document retrieval. MedLinker is trained and evaluated on this subset (st21pv) of MedMentions.

Regarding the concept aliases present in UMLS, we found it useful to introduce some additional restrictions that improve the processing speed of our string matching methods while maintaining task performance. We discard aliases longer than 5 tokens and aliases that include punctuation (except dashes). Aliases are lowercased, along with the query strings used with dictionary methods.

MedMentions uses the PubTator format, which annotates entities at the character-level. Since our methods require annotations at the token-level, we also preprocess the dataset with a tokenizer specialized for the medical domain. We use sciSpacy [7] for tokenization and sentence splitting, which is trained for the biomedical domain. Occasionally, sentence splitting errors incurred in misaligned mentions, and thus missing from training and evaluation. From a total of 203,282 annotations, this step produced 321 misalignments (0.16%).

4 Solution

In this section we describe our methods for mention recognition, entity matching using string-based methods and contextual embeddings, and finally how the different matchers are combined into our final predictions. In Fig. 1 we show how the major components of our solution interact with each other.

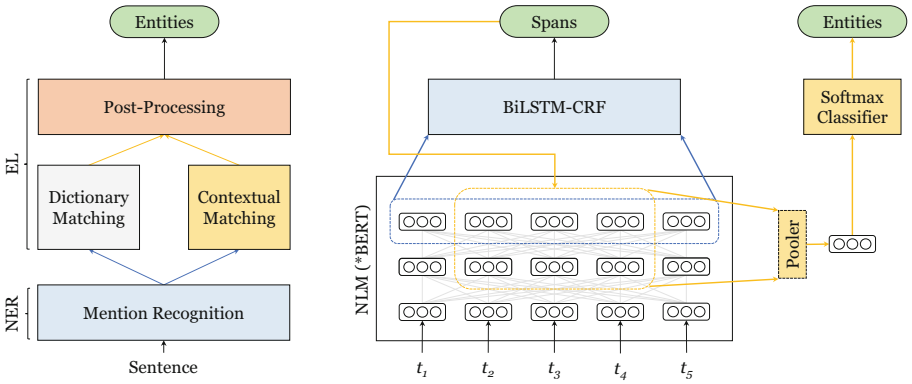

Fig. 1. Left: Overview of our solution, showing NER producing mentions that are matched to entities using independent approaches based on ngrams and contextual embeddings, which are combined in a post-processing step into the final entity predictions. Right: Detailed view into how we use NLMs to first derive spans from NER based on the last states of the NLM, then match entities based on a pooled representation of the predicted span (e.g. states for tokens t_2, t_3 and t_4 at layers $-1, -2$).

Mention Recognition Using NLMs. We follow the standard architecture for neural-based NER, using contextual embeddings from NLMs specialized for the medical domain [2–4]. This architecture is a BiLSTM that handles sequential encoding with long-term dependencies, together with a Conditional Random Field (CRF) which uses the BiLSTM's final states to improve dependencies between output labels. Similarly to [2], this model also employs character-level embeddings, learned during training, to capture morphological information.

Zero-Shot Linking with Approximate Dictionary Matching. Using SimString [10], we represent our restricted set of aliases from UMLS as character n-grams, similarly to previous works [11,12]. After experimenting with different sizes, we found that char n-grams of size 3 performed best. SimString matches strings (i.e. aliases) using the highly scalable CPMerge algorithm which is designed to find similar strings based on overlapping features (i.e. char n-grams).

Given aliases $a \in A_{\text{UMLS}}$, corresponding char n-gram features $\hat{a}$, and a function *map* for mapping entities (concepts) $e \in E_{\text{CUI/STY}}$ to aliases, we match query strings s, with char n-gram features $\hat{s}$, using the scoring function:

$$scoreSTR(s, e) = \max_{a \in map(e)} \cos(\hat{s}, \hat{a}) \quad (1) \qquad \cos(\hat{s}, \hat{a}) = \frac{|\hat{s} \cap \hat{a}|}{\sqrt{|\hat{s}|\,|\hat{a}|}} \quad (2)$$

Linking by Similarity to Entity Embeddings. Considering that MedMentions includes a large set of annotated spans, similarly to WSD corpora, we replicate the pooling method used in [9]. Essentially, we represent STYs and CUIs as the average of all their corresponding contextual embeddings, which are, in turn, represented by the sum of the embeddings from the last 4 layers of the NLM. This results in 21 STY embeddings, and 18,425 CUI embeddings that can be matched using Nearest Neighbors (1NN, only most similar) in NLM-space.

Given entities $e \in E_{\text{CUI/STY}}$, and corresponding precomputed embeddings $\overrightarrow{e}$, we match query strings s, with embedding $\overrightarrow{s}$, obtained by applying the same pooling procedure (with the full sentence), using the scoring function:

$$score1NN(s, e) = \cos(\overrightarrow{s}, \overrightarrow{e}) = \frac{\overrightarrow{s} \cdot \overrightarrow{e}}{\|\overrightarrow{s}\| \, \|\overrightarrow{e}\|} \quad (3)$$

Linking by Classifying Contextual Embeddings. Even though the 1NN approach is very successful for WSD, we also take this opportunity to experiment with training a classifier using contextual embeddings from NLMs, instead of averaging all contextual embeddings grouped by entity. In particular, we train a minimal softmax classifier, without hidden layers (# parameters = # embedding dimensions * # entities), on the annotations of the training set.

Using the same notation defined previously, we match query spans to entities using the following scoring function:

$$scoreCLF(s, e) = P(e|\overrightarrow{s}) = \frac{\exp^{f(\overrightarrow{s})_j}}{\sum_{i=1}^{|E|} \exp^{f(\overrightarrow{s})_i}} \quad (4) \qquad f : \mathbb{R}^{dim(\overrightarrow{s})} \rightarrow \mathbb{R}^{|E|} \quad (5)$$

The function f produces the output vector using weights learned during training with ADAM optimization and categorical cross-entropy loss. The output vector is processed by the softmax function to provide our class (entity) probabilities. We train for 100 epochs, with patience limit of 10, using a batch size of 64.

Combining String and Contextual Matching. We effectively combine our matchers using the following straightforward ensembling method (see Sect. 5):

$$scoreSTR\_CTX(s, e) = \max(scoreSTR(s, e), scoreCTX(s, e)) \quad (6)$$

where $scoreCTX$ can correspond either to $score1NN$ or $scoreCLF$, depending on the configuration being tested. Still, this method exhibits an undesirable bias towards higher recall and lower precision. Therefore, we introduce a post-processing (PP) step to minimize false positives. We train a Logistic Regression Binary Classifier, on the training set, using the following five features:

- $\max_{e \in E} scoreSTR(s, e)$ (max string).
- $\max_{e \in E} scoreCTX(s, e)$ (max context).
- $\max_{e \in E}(scoreSTR(s, e), scoreCTX(s, e))$ (max overall).
- $(\max_{e \in E} scoreSTR(s, e) + \max_{e \in E} scoreCTX(s, e))/2$ (average).
- $\text{argmax}_{e \in E} scoreSTR(s, e) == \text{argmax}_{e \in E} scoreCTX(s, e)$ (agreement).

Testing this classifier with different thresholds, we're able to determine the optimal thresholds for balanced Precision and Recall performance (see Fig. 2).

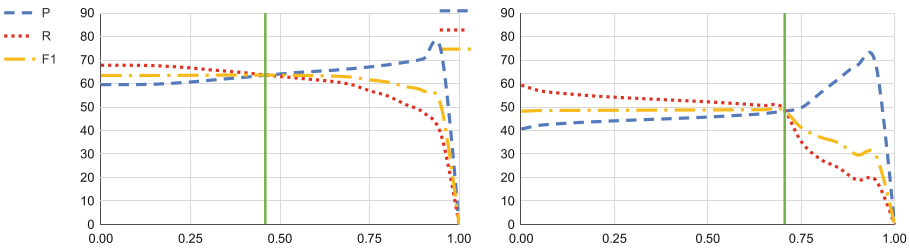

Fig. 2. Performance variation on validation set of MedMentions st21pv (Left: STY, Right: CUI) using different thresholds on the decision filter. Vertical line (green) marks the threshold which resulted in the best balance between Precision and Recall (Color figure online).

5 Evaluation

We evaluate our solution on the test split of MedMentions st21pv, following the metrics described in the MedMentions paper [1]. We require that $scoreSTR$, $score1NN$ and $scoreCLF$ have scores above 0.5 in order to reduce the incidence of false positives. In the event of ties, matches are sorted alphabetically.

Mention Recognition. As described in Sect. 4, our efforts on this task focus on experimenting with different NLMs specialized for the medical domain. On Table 1 we show that the various specialized NLMs we used to initialize the BiLSTM-CRF model produce comparable results on this task.

Table 1. Mention recognition performance using different specialized NLMs.

Model	P	R	F1
Exact match	51.32	32.96	40.14
NCBI BERT (uncased)	**69.44**	69.38	69.41
BioBERT 1.1 (cased)	70.00	70.43	70.21
SciBERT (SciVocab)			
- Uncased	69.42	**71.81**	**70.59**
- Cased	69.16	71.30	70.22

Mention Linking. On Table 2 we present our results for Entity Linking using the predicted spans from the SciBERT (uncased) based NER model that performed best for Mention Recognition. These results show that our solution performs competitively, achieving state-of-the-art on CUI Linking, and comparable results to the state-of-the-art on STY Linking. Additionally, we also report performance using the different scoring functions covered in this paper. $score1NN$ outperforms $scoreCLF$ on CUI Linking, by a small margin, but on STY Linking $scoreCLF$ substantially outperforms $score1NN$. We believe these differences can be explained from the fact that all STYs are represented in the training set, while the overlap between CUIs in the training and test sets is low.

Category Performance. Using gold spans to focus on linking performance, we notice[3] that types/categories such as 'T005-Virus' obtain 88 F1, while 'T022-Body System' obtains only 44 F1, on STY Linking. CUI Linking results show similar variations, although with a stronger tendency towards better performance on concepts belonging to narrower types (i.e. STYs encompassing fewer CUIs).

[3] Results for all categories: https://github.com/danlou/medlinker/categories.pdf.

Table 2. Semantic Type (STY) and Concept (CUI) Linking performance comparison. † were produced using the same st21pv subset of UMLS release 2017 AA Active.

Model	STY Linking			CUI Linking		
	P	R	F1	P	R	F1
Exact match	49.04	31.97	38.71	47.12	31.11	37.48
QuickUMLS† (v1.3) [11]	14.51	16.87	15.60	17.98	26.11	21.30
ScispaCy† (v0.2.4) [7]	10.14	31.68	15.36	25.17	53.52	34.24
TaggerOne [1]	N/A	N/A	N/A	47.10	43.60	45.30
Nejadgholi et al. [5]						
- BioBERT	61	66	63	N/A	N/A	N/A
- BioBERT|BERT-base	**63**	65	**64**	N/A	N/A	N/A
MedLinker						
- $scoreSTR$	48.31	56.81	52.22	33.03	47.34	38.91
- $score1NN$	46.62	62.67	53.47	33.61	55.16	41.77
- $scoreCLF$	58.62	64.63	61.48	32.21	52.66	39.97
- $scoreSTR\_1NN$	53.06	65.94	58.80	40.46	**59.69**	48.23
- $scoreSTR\_CLF$	59.23	**67.81**	63.23	40.70	59.59	48.37
- $scoreSTR\_CLF$ (PP, bal. thresh.)	**63.13**	63.69	63.41	**48.43**	50.07	**49.24**

6 Conclusion

A major issue in biomedical NLP or IR research is the fact that annotated datasets are scarce and expensive to collect. While that problem is unlikely to improve in the near future, this work has shown that there's still significant room for improvement by simply adapting existing approaches, such as end-to-end neural models for NER or WSD, making them easier to integrate with previous works often unassociated to those approaches, such as ADM. From a more practical perspective, this work pushes the state-of-the-art on the new and challenging MedMentions dataset, while using a modular approach that can be further improved with the integration of additional IR methods in future work.

References

1. Mohan, S., Li, D.: MedMentions: a large biomedical corpus annotated with UMLS concepts. In: AKBC (2019)
2. Beltagy, I., Lo, K., Cohan, A.: SciBERT: a pretrained language model for scientific text. In: IJCNLP (2019)
3. Lee, J., et al.: BioBERT: a pre-trained biomedical language representation model for biomedical text mining. Bioinformatics **36**(4), 1234–1240 (2020)
4. Peng, Y., Yan, S., Lu, Z.: Transfer learning in biomedical natural language processing: an evaluation of BERT and ELMo on ten benchmarking datasets. In: BioNLP@ACL (2019)

5. Nejadgholi, I., Fraser, K.C., De Bruijn, B., Li, M., LaPlante, A., Abidine, K.Z.: Extracting UMLS concepts from medical text using general and domain-specific deep learning models. In: LOUHI@EMNLP (2019)
6. Bodenreider, O.: The unified medical language system (UMLS): integrating biomedical terminology. Nucleic Acids Res. **32**(suppl_1), 267–270 (2004)
7. Neumann, M., King, D., Beltagy, I., Ammar, W.: ScispaCy: fast and robust models for biomedical natural language processing. In: BioNLP@ACL (2019)
8. Leaman, R., Lu, Z.: TaggerOne: joint named entity recognition and normalization with semi-Markov models. Bioinformatics **32**(18), 2839–2846 (2016)
9. Loureiro, D., Jorge, A.M.: Language modelling makes sense: propagating representations through wordnet for full-coverage word sense disambiguation. In: ACL (2019)
10. Okazaki, N., Tsujii, J.: Simple and efficient algorithm for approximate dictionary matching. In: COLING (2010)
11. Soldaini, L., Goharian, N.: QuickUMLS: a fast, unsupervised approach for medical concept extraction. In: MedIR Workshop, SIGIR (2016)
12. Kaewphan, S., Hakala, K., Miekka, N., Salakoski, T., Ginter, F.: Wide-scope biomedical named entity recognition and normalization with CRFs, fuzzy matching and character level modeling. Database **2018**, (2018)